→ Convenciones:

```
# En todos los nodos como 'sudo su'.
[root@srv1 ~]# Solo en servidor 'srv1'→ como 'sudo su'.
[root@srv2 ~]# Solo en servidor 'srv2'→ como 'sudo su'.
```

364.2 Advanced RAID (weight: 2)

Weight	2
Description	Candidates should be able to manage software raid devices on Linux. This includes advanced features such as partitonable RAIDs and RAID containers as well as recovering RAID arrays after a failure.

Key Knowledge Areas:

- Manage RAID devices using various raid levels, including hot spare discs, partitionable RAIDs and RAID containers
- Add and remove devices from an existing RAID
- Change the RAID level of an existing device
- Recover a RAID device after a failure
- Understand various metadata formats and RAID geometries
- Understand availability and performance properties of various raid levels
- Configure mdadm monitoring and reporting

Partial list of the used files, terms and utilities:

- mdadm
- /proc/mdstat
- /proc/sys/dev/raid/*

→ Conceptos.

RAID es un acrónimo del inglés que significa **Redundant Array of Independent Disks**, literalmente «matriz de discos independientes redundantes», aunque no todos los sistemas RAID proporcionan redundancia.

En otras palabras, consiste en crear un único volumen con varios discos duros funcionando en conjunto, y con este conjunto se puede conseguir redundancia (tolerancia a fallos en el caso de que uno falle, conocido como **disk mirroring** o mayor velocidad (conocido como **disk striping**), haciendo que ese conjunto sea en realidad un tándem.

Niveles RAID estándar

Los niveles RAID más comúnmente usados son:

- RAID 0: Conjunto dividido

- RAID 1: Conjunto en espejo
- RAID 5: Conjunto dividido con paridad distribuida

RAID 0 (Data Striping, Striped Volume)

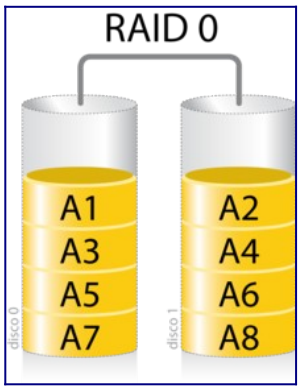


Diagrama de una configuración RAID 0

Un RAID 0 (también llamado conjunto dividido, volumen dividido, volumen seccionado) distribuye los datos equitativamente entre dos o más discos (usualmente se ocupa el mismo espacio en dos o más discos) sin información de paridad que proporcione redundancia. Es importante señalar que el RAID 0 no era uno de los niveles RAID originales y que no es redundante. El RAID 0 se usa habitualmente para proporcionar un alto rendimiento de escritura ya que los datos se escriben en dos o más discos de forma paralela, aunque un mismo fichero solo está presente una vez en el conjunto. RAID 0 también puede utilizarse como forma de crear un pequeño número de grandes discos virtuales a partir de un gran número de pequeños discos físicos. Un RAID 0 puede ser creado con discos de diferentes tamaños, pero el espacio de almacenamiento añadido al conjunto estará limitado por el tamaño del disco más pequeño (por ejemplo, si se hace un conjunto dividido con un disco de 450 GB y otro de 100 GB, el tamaño del conjunto resultante será solo de 200 GB, ya que cada disco aporta 100 GB). Una buena implementación de un RAID 0 dividirá las operaciones de lectura y escritura en bloques de igual tamaño, por lo que distribuirá la información equitativamente entre los dos discos. También es posible crear un RAID 0 con más de dos discos, si bien, la fiabilidad del conjunto será igual a la fiabilidad media de cada disco entre el número de discos del conjunto; es decir, la fiabilidad total —medida como MTTF o MTBF— es (aproximadamente) inversamente proporcional al número de discos del conjunto (pues para que el conjunto falle es suficiente con que lo haga *cualquiera* de sus discos). No debe confundirse RAID 0 con un Volumen Distribuido (Spanned Volume) en el cual se agregan múltiples espacios no usados de varios discos para formar un único disco virtual. Es posible que en un Volumen Distribuido el fichero a recuperar esté presente en un solo disco del conjunto, debido a que aquí no hay una distribución equitativa de los datos (como se mencionó, para RAID 0); por lo tanto, en ese caso no sería posible la recuperación paralela de datos y no mejoraría el rendimiento de lectura.

RAID 1 (espejo)

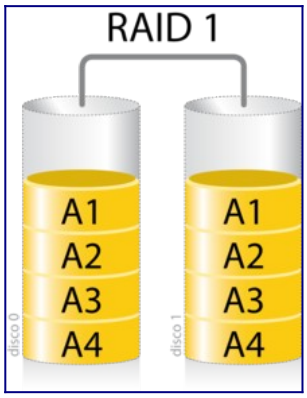


Diagrama de una configuración RAID 1

Un RAID 1 crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos. Esto resulta útil cuando queremos tener más seguridad desaprovechando capacidad, ya que si perdemos un disco, tenemos el otro con la misma información. Un conjunto RAID 1 solo puede ser tan grande como el más pequeño de sus discos. Un RAID 1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un solo disco; es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan *todos* sus discos).

Además, dado que todos los datos están en dos o más discos, con hardware habitualmente independiente, el rendimiento de lectura se incrementa aproximadamente como múltiplo lineal del número de copias; es decir, un RAID 1 puede estar leyendo simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica. Para maximizar los beneficios sobre el rendimiento del RAID 1 se recomienda el uso de controladoras de disco independientes, una para cada disco (práctica que algunos denominan *splitting* o *duplexing*).

Como en el RAID 0, el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo la tasa de transferencia, con el único límite de la velocidad soportada por la controladora RAID. Sin embargo, muchas tarjetas RAID 1 IDE antiguas leen solo de un disco de la pareja, por lo que su rendimiento es igual al de un único disco. Algunas implementaciones RAID 1 antiguas también leen de ambos discos simultáneamente y comparan los datos para detectar errores.

Al escribir, el conjunto se comporta como un único disco, dado que los datos deben ser escritos en todos los discos del RAID 1. Por tanto, el rendimiento de escritura no mejora.

El RAID 1 tiene muchas ventajas de administración. Por ejemplo, en algunos entornos 24/7, es posible «dividir el espejo»: marcar un disco como inactivo, hacer una copia de seguridad de dicho disco y luego «reconstruir» el espejo. Esto requiere que la aplicación de gestión del conjunto soporte la recuperación de los datos del disco en el momento de la división. Este procedimiento es menos crítico que la presencia de una característica de *snapshot* en algunos sistemas de archivos, en

la que se reserva algún espacio para los cambios, presentando una vista estática en un punto temporal dado del sistema de archivos. Alternativamente, un conjunto de discos puede ser almacenado de forma parecida a como se hace con las tradicionales cintas.

RAID 2

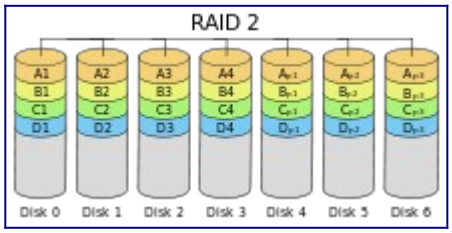


Diagrama de una configuración RAID 2

Distribuye los datos entrelazadas a nivel de bit. El código de error se intercala a través de varios discos también a nivel de bit, el código de error se calcula con el código de Hamming. Todo giro del cabezal de disco se sincroniza y los datos se distribuyen en bandas de modo que cada bit secuencial está en una unidad diferente. La paridad de Hamming se calcula a través de y los bits correspondientes y se almacena en al menos un disco de paridad. Este nivel es solo significante a nivel histórico y teórico, ya que actualmente no se utiliza.

RAID 3

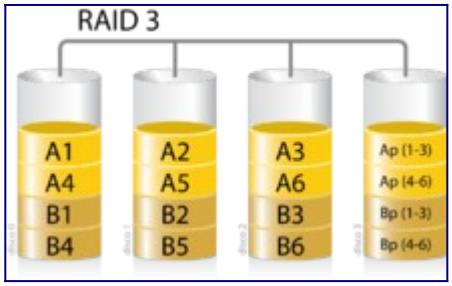


Diagrama de una configuración RAID 3. Cada número representa un byte de datos; cada columna, un disco.

Un RAID 3 divide los datos a nivel de bytes en lugar de a nivel de bloques . Los discos son sincronizados por la controladora para funcionar al unísono. Este es el único nivel RAID original que actualmente no se usa. Permite tasas de transferencias extremadamente altas. Un RAID 3 necesitaría un mínimo de tres discos, utilizando uno para datos de paridad. En estos se copian los datos en distribución RAID 0 en los 2 primeros discos, sin embargo, en el tercer disco, se crea el byte de paridad. Esto quiere decir que si por ejemplo perdemos un byte de uno de los discos, siempre podremos recuperarlo mediante el byte de paridad que se ha generado anteriormente.

En el ejemplo del gráfico, una petición del bloque «A56» formado por los bytes Ah1 a Af6 requeriría que los tres discos de datos buscaran el comienzo (Ag1) y devolvieran su contenido. Una

petición simultánea del bloque «Bh» en el cual guarda la suma de los números de un archivo y en caso de pérdida de datos se hace la diferencia con la suma o la multiplicación incluso.

RAID 4

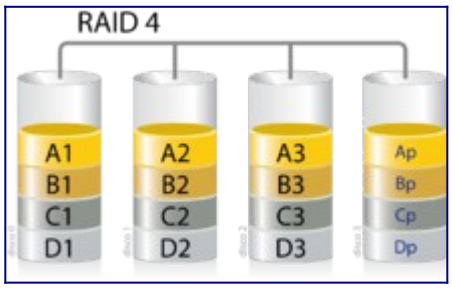


Diagrama de una configuración RAID 4. Cada número representa un bloque de datos; cada columna, un disco.

Un RAID 4, también conocido como IDA (acceso independiente con discos dedicados a la paridad), usa división a nivel de bloques con un disco de paridad dedicado. Necesita un mínimo de 3 discos físicos. El RAID 4 es parecido al RAID 3 excepto porque divide a nivel de bloques en lugar de a nivel de bytes. Esto permite que cada miembro del conjunto funcione independientemente cuando se solicita un único bloque. Si la controladora de disco lo permite, un conjunto RAID 4 puede servir varias peticiones de lectura simultáneamente. En principio también sería posible servir varias peticiones de escritura simultáneamente, pero al estar toda la información de paridad en un solo disco, este se convertiría en el cuello de botella del conjunto.

En el gráfico de ejemplo anterior, una petición del bloque «A1» sería servida por el disco 0. Una petición simultánea del bloque «B1» tendría que esperar, pero una petición de «B2» podría atenderse concurrentemente.

RAID 5

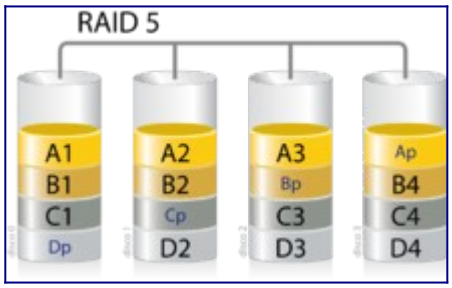


Diagrama de una configuración RAID 5

Un RAID 5 (también llamado distribuido con paridad) es una división de datos a nivel de bloques que distribuye la información de paridad entre todos los discos miembros del conjunto. El RAID 5 ha logrado popularidad gracias a su bajo coste de redundancia. Generalmente, el RAID 5 se implementa con soporte hardware para el cálculo de la paridad. RAID 5 necesitará un mínimo de 3

discos para ser implementado.

En el gráfico de ejemplo anterior, una petición de lectura del bloque «A1» sería servida por el disco 0. Una petición de lectura simultánea del bloque «B1» tendría que esperar, pero una petición de lectura de «B2» podría atenderse concurrentemente ya que sería servida por el disco 1.

Cada vez que un bloque de datos se escribe en un RAID 5, se genera un bloque de paridad dentro de la misma división (*stripe*). Un bloque se compone a menudo de muchos sectores consecutivos de disco. Una serie de bloques (un bloque de cada uno de los discos del conjunto) recibe el nombre colectivo de división (*stripe*). Si otro bloque, o alguna porción de un bloque, es escrita en esa misma división, el bloque de paridad (o una parte del mismo) es recalculada y vuelta a escribir. El disco utilizado por el bloque de paridad está escalonado de una división a la siguiente, de ahí el término «bloques de paridad distribuidos». Las escrituras en un RAID 5 son costosas en términos de operaciones de disco y tráfico entre los discos y la controladora.

Los bloques de paridad no se leen en las operaciones de lectura de datos, ya que esto sería una sobrecarga innecesaria y disminuiría el rendimiento. Sin embargo, los bloques de paridad se leen cuando la lectura de un sector de datos provoca un error de CRC. En este caso, el sector en la misma posición relativa dentro de cada uno de los bloques de datos restantes en la división y dentro del bloque de paridad en la división se utilizan para reconstruir el sector erróneo. El error CRC se oculta así al resto del sistema. De la misma forma, si falla un disco del conjunto, los bloques de paridad de los restantes discos son combinados matemáticamente con los bloques de datos de los restantes discos para reconstruir los datos del disco que ha fallado «al vuelo».

Lo anterior se denomina a veces Modo Interino de Recuperación de Datos (*Interim Data Recovery Mode*). El sistema sabe que un disco ha fallado, pero solo con el fin de que el sistema operativo pueda notificar al administrador que una unidad necesita ser reemplazada: las aplicaciones en ejecución siguen funcionando ajenas al fallo. Las lecturas y escrituras continúan normalmente en el conjunto de discos, aunque con alguna degradación de rendimiento. La diferencia entre el RAID 4 y el RAID 5 es que, en el Modo Interno de Recuperación de Datos, el RAID 5 puede ser ligeramente más rápido, debido a que, cuando el CRC y la paridad están en el disco que falló, los cálculos no tienen que realizarse, mientras que en el RAID 4, si uno de los discos de datos falla, los cálculos tienen que ser realizados en cada acceso.

El fallo de un segundo disco provoca la pérdida completa de los datos.

El número máximo de discos en un grupo de redundancia RAID 5 es teóricamente ilimitado, pero en la práctica es común limitar el número de unidades. Los inconvenientes de usar grupos de redundancia mayores son una mayor probabilidad de fallo simultáneo de dos discos, un mayor tiempo de reconstrucción y una mayor probabilidad de hallar un sector irrecuperable durante una reconstrucción. A medida que el número de discos en un conjunto RAID 5 crece, el MTBF (tiempo medio entre fallos) puede ser más bajo que el de un único disco. Esto sucede cuando la probabilidad de que falle un segundo disco en los N-1 discos restantes de un conjunto en el que ha fallado un

disco en el tiempo necesario para detectar, reemplazar y recrear dicho disco es mayor que la probabilidad de fallo de un único disco. Una alternativa que proporciona una protección de paridad dual, permitiendo así mayor número de discos por grupo, es el RAID 6.

Algunos vendedores RAID evitan montar discos de los mismos lotes en un grupo de redundancia para minimizar la probabilidad de fallos simultáneos al principio y el final de su vida útil.

Las implementaciones RAID 5 presentan un rendimiento malo cuando se someten a cargas de trabajo que incluyen muchas escrituras más pequeñas que el tamaño de una división (*stripe*). Esto se debe a que la paridad debe ser actualizada para cada escritura, lo que exige realizar secuencias de lectura, modificación y escritura tanto para el bloque de datos como para el de paridad. Implementaciones más complejas incluyen a menudo cachés de escritura no volátiles para reducir este problema de rendimiento.

En el caso de un fallo del sistema cuando hay escrituras activas, la paridad de una división (*stripe*) puede quedar en un estado inconsistente con los datos. Si esto no se detecta y repara antes de que un disco o bloque falle, pueden perderse datos debido a que se usará una paridad incorrecta para reconstruir el bloque perdido en dicha división. Esta potencial vulnerabilidad se conoce a veces como «agujero de escritura». Son comunes el uso de caché no volátiles y otras técnicas para reducir la probabilidad de ocurrencia de esta vulnerabilidad.

RAID 6

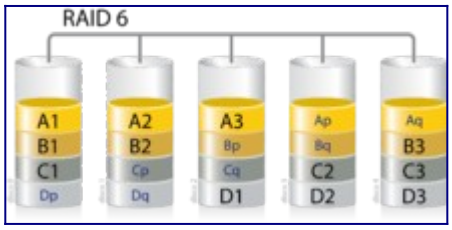


Diagrama de una configuración RAID 6. Cada número representa un bloque de datos; cada columna, un disco; *p* y *q*, códigos Reed-Solomon.

Un RAID 6 amplía el nivel RAID 5 añadiendo otro bloque de paridad, por lo que divide los datos a nivel de bloques y distribuye los dos bloques de paridad entre todos los miembros del conjunto. El RAID 6 no era uno de los niveles RAID originales.

El RAID 6 puede ser considerado un caso especial de código Reed-Solomon.2 El RAID 6, como es un caso degenerado, exige solo sumas en el Campo de galois. Dado que se está operando sobre bits, lo que se usa es un campo binario de Galois ( ). En las representaciones cíclicas de los campos binarios de Galois, la suma se calcula con un simple XOR.

Tras comprender el RAID 6 como caso especial de un código Reed-Solomon, se puede ver que es posible ampliar este enfoque para generar redundancia simplemente produciendo otro código, típicamente un polinomio en (m = 8 significa que estamos operando sobre bytes). Al añadir

códigos adicionales es posible alcanzar cualquier número de discos redundantes, y recuperarse de un fallo de ese mismo número de discos en cualquier punto del conjunto, pero en el nivel RAID 6 se usan dos únicos códigos.

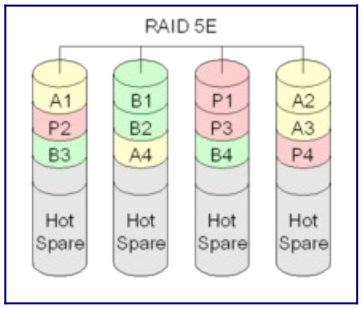
Al igual que en el RAID 5, en el RAID 6 la paridad se distribuye en divisiones (*stripes*), con los bloques de paridad en un lugar diferente en cada división.

El RAID 6 es muy eficaz cuando se usa un pequeño número de discos, pero a medida que el conjunto crece y se dispone de más discos la pérdida en capacidad de almacenamiento se hace menos importante, creciendo al mismo tiempo la probabilidad de que dos discos fallen simultáneamente. El RAID 6 proporciona protección contra fallos dobles de discos y contra fallos cuando se está reconstruyendo un disco. En caso de que solo tengamos un conjunto puede ser más adecuado que usar un RAID 5 con un disco de reserva (*hot spare*).

La capacidad de datos de un conjunto RAID 6 es *n*-2, y *n* es el número total de discos del conjunto.

Un RAID 6 no penaliza el rendimiento de las operaciones de lectura, pero sí el de las de escritura debido al proceso que exigen los cálculos adicionales de paridad. Esta penalización puede minimizarse agrupando las escrituras en el menor número posible de divisiones (*stripes*), lo que puede lograrse mediante el uso de un sistema de archivos WAFL.

RAID 5E y RAID 6E



RAID 5E

Se puede llamar RAID 5E y RAID 6E a las variantes de RAID 5 y RAID 6 que incluyen discos de reserva. Estos discos pueden estar conectados y preparados (*hot spare*) o en espera (*standby spare*). En los RAID 5E y RAID 6E, los discos de reserva están disponibles para cualquiera de las unidades miembro. No suponen mejora alguna del rendimiento, pero sí se minimiza el tiempo de reconstrucción (en el caso de los discos *hot spare*) y las labores de administración cuando se producen fallos. Un disco de reserva no es realmente parte del conjunto hasta que un disco falla y el conjunto se reconstruye sobre el de reserva.

Niveles RAID anidados (de nidos)

Muchas controladoras permiten anidar niveles RAID, es decir, que un RAID pueda usarse como elemento básico de otro en lugar de discos físicos. Resulta instructivo pensar en estos conjuntos

como capas dispuestas unas sobre otras, con los discos físicos en la inferior.

Los RAID anidados se indican normalmente uniendo en un solo número los correspondientes a los niveles RAID usados, añadiendo a veces un «+» entre ellos. Por ejemplo, el RAID 10 (o RAID 1+0) consiste conceptualmente en múltiples conjuntos de nivel 1 almacenados en discos físicos con un nivel 0 encima, agrupando los anteriores niveles 1. En el caso del RAID 0+1 se usa más esta forma que RAID 01 para evitar la confusión con el RAID 1. Sin embargo, cuando el conjunto de más alto nivel es un RAID 0 (como en el RAID 10 y en el RAID 50), la mayoría de los vendedores eligen omitir el «+», a pesar de que RAID 5+0 sea más informativo.

Al anidar niveles RAID, se suele combinar un nivel RAID que proporcione redundancia con un RAID 0 que aumenta el rendimiento. Con estas configuraciones es preferible tener el RAID 0 como nivel más alto y los conjuntos redundantes debajo, porque así será necesario reconstruir menos discos cuando uno falle. (Así, el RAID 10 es preferible al RAID 0+1 aunque las ventajas administrativas de «dividir el espejo» del RAID 1 se perderían.)

Los niveles RAID anidados más comúnmente usados son:

- RAID 0+1: Un espejo de divisiones
- RAID 1+0: Una división de espejos
- RAID 30: Una división de niveles RAID con paridad dedicada
- RAID 100: Una división de una división de espejos
- RAID 10+1: Un Espejo de espejos

RAID 0+1

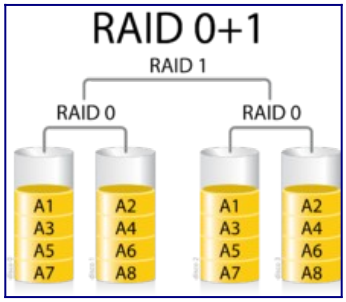


Diagrama de una configuración RAID 0+1

Un **RAID 0+1** (también llamado **RAID 01**, que no debe confundirse con RAID 1) es un RAID usado para replicar y compartir datos entre varios discos. La diferencia entre un RAID 0+1 y un RAID 1+0 es la localización de cada nivel RAID dentro del conjunto final: un RAID 0+1 es un espejo de divisiones.3

Como puede verse en el diagrama, primero se crean dos conjuntos RAID 0 (dividiendo los datos en discos) y luego, sobre los anteriores, se crea un conjunto RAID 1 (realizando un espejo de los anteriores). La ventaja de un RAID 0+1 es que cuando un disco duro falla, los datos perdidos

pueden ser copiados del otro conjunto de nivel 0 para reconstruir el conjunto global. Sin embargo, añadir un disco duro adicional en una división, es obligatorio añadir otro al de la otra división para equilibrar el tamaño del conjunto.

Además, el RAID 0+1 no es tan robusto como un RAID 1+0, no pudiendo tolerar dos fallos simultáneos de discos salvo que sean en la misma división. Es decir, cuando un disco falla, la otra división se convierte en un punto de fallo único. Además, cuando se sustituye el disco que falló, se necesita que todos los discos del conjunto participen en la reconstrucción de los datos.

Con la cada vez mayor capacidad de las unidades de discos (liderada por las unidades serial ATA), el riesgo de fallo de los discos es cada vez mayor. Además, las tecnologías de corrección de errores de bit no han sido capaces de mantener el ritmo de rápido incremento de las capacidades de los discos, provocando un mayor riesgo de hallar errores físicos irrecuperables.

Dado esto cada vez tiene mayores riesgos el RAID 0+1 (y su vulnerabilidad ante los fallos dobles simultáneos), muchos entornos empresariales críticos están empezando a evaluar configuraciones RAID más tolerantes de fallos que añaden un mecanismo de paridad subyacente. Entre los más prometedores están los enfoques híbridos como el RAID 0+1+5 (espejo sobre paridad única) o RAID 0+1+6 (espejo sobre paridad dual). Son los más habituales por las empresas.

RAID 1+0

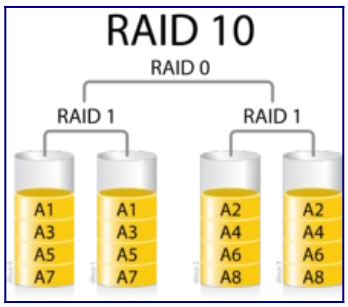


Diagrama de una configuración RAID 10

Un **RAID 1+0**, a veces llamado **RAID 10**, es lo más parecido a un RAID 0+1 con la excepción de que los niveles RAID que lo forman se invierte: el RAID 10 es una división de espejos.

En cada división RAID 10 o RAID 1+0, pueden fallar todos los discos salvo uno sin que se pierdan datos. Sin embargo, si los discos que han fallado no se reemplazan, el restante pasa a ser un punto único de fallo para todo el conjunto. Si ese disco falla entonces, se perderán todos los datos del conjunto completo. Como en el caso del RAID 0+1, si un disco que ha fallado no se reemplaza, entonces un solo error de medio irrecuperable que ocurra en el disco espejado resultaría en pérdida de datos.

Debido a estos mayores riesgos del RAID 1+0, muchos entornos empresariales críticos están empezando a evaluar configuraciones RAID más tolerantes de fallos que añaden un mecanismo de

paridad subyacente. Entre los más prometedores están los enfoques híbridos como el RAID 0+1+5 (espejo sobre paridad única) o RAID 0+1+6 (espejo sobre paridad dual).

El RAID 10 es a menudo la mejor elección para bases de datos de altas prestaciones, debido a que la ausencia de cálculos de paridad proporciona mayor velocidad de escritura.

RAID 30

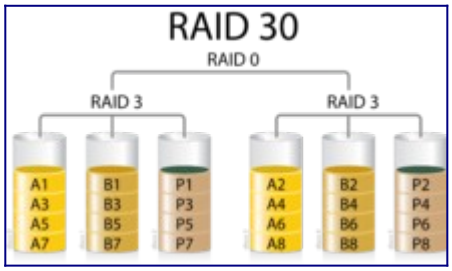
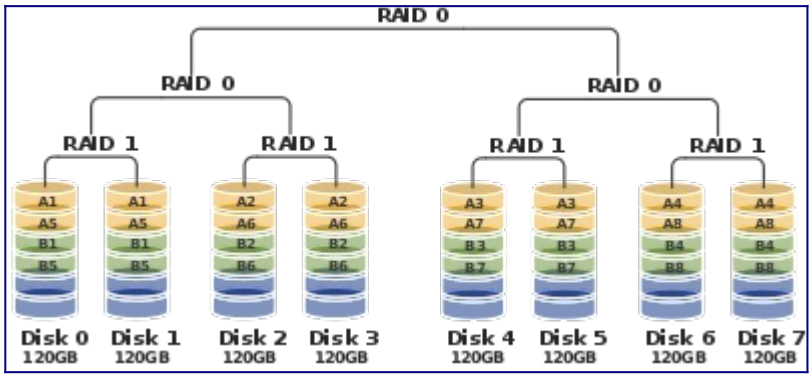


Diagrama de una configuración RAID 30

El **RAID 30** o división con conjunto de paridad dedicado es una combinación de un RAID 3 y un RAID 0. El RAID 30 proporciona tasas de transferencia elevadas combinadas con una alta fiabilidad a cambio de un coste de implementación muy alto. La mejor forma de construir un RAID 30 es combinar dos conjuntos RAID 3 con los datos divididos en ambos conjuntos. El RAID 30 trocea los datos en bloques más pequeños y los divide en cada conjunto RAID 3, que a su vez lo divide en trozos aún menores, calcula la paridad aplicando un XOR a cada uno y los escriben en todos los discos del conjunto salvo en uno, donde se almacena la información de paridad. El tamaño de cada bloque se decide en el momento de construir el RAID.Etc...

El RAID 30 permite que falle un disco de cada conjunto RAID 3. Hasta que estos discos que fallaron sean reemplazados, los otros discos de cada conjunto que sufrió el fallo son puntos únicos de fallo para el conjunto RAID 30 completo. En otras palabras, si alguno de ellos falla se perderán todos los datos del conjunto. El tiempo de recuperación necesario (detectar y responder al fallo del disco y reconstruir el conjunto sobre el disco nuevo) representa un periodo de vulnerabilidad para el RAID.

RAID 100



RAID 100

Un **RAID 100**, a veces llamado también **RAID 10+0**, es una división de conjuntos RAID 10. El RAID 100 es un ejemplo de «RAID cuadriculado», un RAID en el que conjuntos divididos son a su vez divididos conjuntamente de nuevo.

Todos los discos menos uno podrían fallar en cada RAID 1 sin perder datos. Sin embargo, el disco restante de un RAID 1 se convierte así en un punto único de fallo para el conjunto degradado. A menudo el nivel superior de división se hace por software. Algunos vendedores llaman a este nivel más alto un *MetaLun* o *Soft Stripe*.

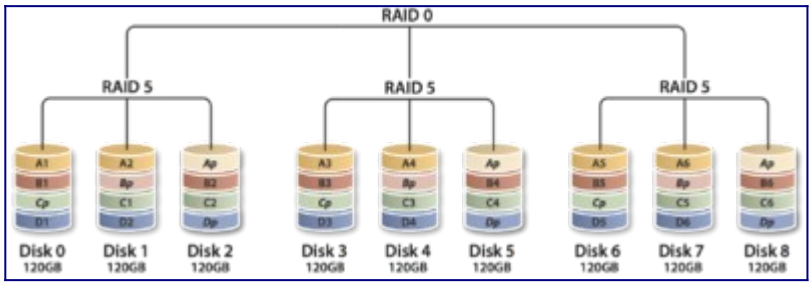
Los principales beneficios de un RAID 100 (y de los RAID cuadriculados en general) sobre un único nivel RAID son mejor rendimiento para lecturas aleatorias y la mitigación de los puntos calientes de riesgo en el conjunto. Por estas razones, el RAID 100 es a menudo la mejor elección para bases de datos muy grandes, donde el conjunto software subyacente limita la cantidad de discos físicos permitidos en cada conjunto estándar. Implementar niveles RAID anidados permite eliminar virtualmente el límite de unidades físicas en un único volumen lógico.

RAID 10+1

Un **RAID 10+1** es un reflejo de dos RAID 10. Se utiliza en la llamados **Network RAID** que aceptan algunas cabinas de datos. Es un sistema de alta disponibilidad por red, lo que permite la replicación de datos entre cabinas a nivel de RAID, con lo cual se simplifica ampliamente la gestión de replicación de cabinas.

El RAID 10+1, tratándose de espejos de RAID10 que tienen una gran velocidad de acceso, hace que el rendimiento sea muy aceptable, siempre y cuando se respete el requerimiento de 2ms de latencia como máximo.

RAID 50



RAID 50

Un **RAID 50**, a veces llamado también **RAID 5+0**, combina la división a nivel de bloques de un RAID 0 con la paridad distribuida de un RAID 5, siendo pues un conjunto RAID 0 dividido de elementos RAID 5.

Un disco de cada conjunto RAID 5 puede fallar sin que se pierdan datos. Sin embargo, si el disco que falla no se reemplaza, los discos restantes de dicho conjunto se convierten en un punto único de fallo para todo el conjunto. Si uno falla, todos los datos del conjunto global se pierden. El tiempo necesario para recuperar (detectar y responder al fallo de disco y reconstruir el conjunto sobre el nuevo disco) representa un periodo de vulnerabilidad del conjunto RAID.

La configuración de los conjuntos RAID repercute sobre la tolerancia a fallos general. Una configuración de tres conjuntos RAID 5 de siete discos cada uno tiene la mayor capacidad y eficiencia de almacenamiento, pero solo puede tolerar un máximo de tres fallos potenciales de disco. Debido a que la fiabilidad del sistema depende del rápido reemplazo de los discos averiados para que el conjunto pueda reconstruirse, es común construir conjuntos RAID 5 de seis discos con un disco de reserva en línea (*hot spare*) que permite empezar de inmediato la reconstrucción en caso de fallo del conjunto. Esto no soluciona el problema de que el conjunto sufre un estrés máximo durante la reconstrucción dado que es necesario leer cada bit, justo cuando es más vulnerable. Una configuración de siete conjuntos RAID 5 de tres discos cada uno puede tolerar hasta siete fallos de disco pero tiene menor capacidad y eficiencia de almacenamiento.

El RAID 50 mejora el rendimiento del RAID 5, especialmente en escritura, y proporciona mejor tolerancia a fallos que un nivel RAID único. Este nivel se recomienda para aplicaciones que necesitan gran tolerancia a fallos, capacidad y rendimiento de búsqueda aleatoria.

A medida que el número de unidades del conjunto RAID 50 crece y la capacidad de los discos aumenta, el tiempo de recuperación lo hace también.

Niveles RAID propietarios

Aunque todas las implementaciones de RAID difieren en algún grado de la especificación idealizada, algunas compañías han desarrollado implementaciones RAID completamente propietarias que difieren sustancialmente de todas las demás.

RAID 50EE

Himperia utiliza el **RAID 50EE** en el ZStore 3212L.4 Se trata de un RAID 0 de dos *pools*, cada uno de ellos con RAID 5EE (7+1+1). Tolera el fallo simultáneo de dos discos, y hasta 4 discos no simultáneos. El tiempo de reconstrucción se reduce al mínimo, gracias al RAID 5EE. Y se mejora el rendimiento gracias al RAID 0.

Paridad doble

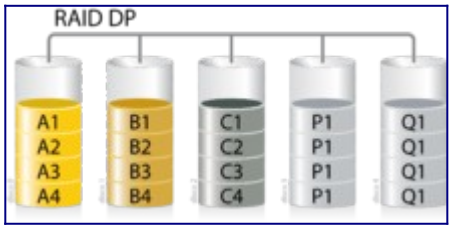


Diagrama una configuración RAID de doble paridad

Una adición frecuente a los niveles RAID existentes es la paridad doble, a veces implementada y conocida como paridad diagonal.5 Como en el RAID 6, hay dos conjuntos de información de chequeo de paridad, pero a diferencia de aquel, el segundo conjunto no es otro conjunto de puntos calculado sobre un síndrome polinomial diferente para los mismos grupos de bloques de datos, sino que se calcula la paridad extra a partir de un grupo diferente de bloques de datos. Por ejemplo, sobre el gráfico tanto el RAID 5 como el RAID 6 calcularían la paridad sobre todos los bloques de la letra A para generar uno o dos bloques de paridad. Sin embargo, es bastante fácil calcular la paridad contra múltiples grupos de bloques, en lugar de solo sobre los bloques de la letra A: puede calcularse la paridad sobre los bloques de la letra A *y* un grupo permutado de bloques.

De nuevo sobre el ejemplo, los bloques Q son los de la paridad doble. El bloque Q2 se calcularía como A2 *xor* B3 *xor* P3, mientras el bloque Q3 se calcularía como A3 *xor* P2 *xor* C1 y el Q1 sería A1 *xor* B2 *xor* C3. Debido a que los bloques de paridad doble se distribuyen correctamente, es posible reconstruir dos discos de datos que fallen mediante recuperación iterativa. Por ejemplo, B2 podría recuperarse sin usar ninguno de los bloque x1 ni x2 mediante el cálculo de B3 *xor* P3 *xor* Q2 = A2, luego A2 *xor* A3 *xor* P1 = A1, y finalmente A1 *xor* C3 *xor* Q1 = B2.

No es recomendable que el sistema de paridad doble funcione en modo degradado debido a su bajo rendimiento.

RAID 1.5

RAID 1.5 es un nivel RAID propietario de HighPoint a veces incorrectamente denominado RAID 15. Por la poca información disponible, parece ser una implementación correcta de un RAID 1. Cuando se lee, los datos se recuperan de ambos discos simultáneamente y la mayoría del trabajo se hace en hardware en lugar de en el controlador software.

RAID 15 se compone de al menos tres elementos lógicos (el requisito mínimo para RAID 5) que son a su vez compuesta de matrices RAID 1. RAID 51 es exactamente lo contrario: que refleja dos matrices RAID 5.

No es difícil ver que la combinación de dos modos RAID mejora en gran medida la seguridad de datos. Con una matriz RAID 15, una unidad puede fallar en cada bloque RAID 1 sin poner todo el sistema al borde del desastre.

RAID 7

RAID 7 es una marca registrada de Storage Computer Corporation, que añade cachés a un RAID 3 o RAID 4 para mejorar el rendimiento.

RAID S o RAID de paridad

RAID S es un sistema RAID de paridad distribuida propietario de EMC Corporation usado en sus sistemas de almacenamiento Symmetrix. Cada volumen reside en un único disco físico, y se combinan arbitrariamente varios volúmenes para el cálculo de paridad. EMC llamaba originalmente a esta característica RAID S y luego la rebautizó **RAID de paridad** (*Parity RAID*) para su plataforma Symmetrix DMX. EMC ofrece también actualmente un RAID 5 estándar para el Symmetrix DMX.

Matrix RAID

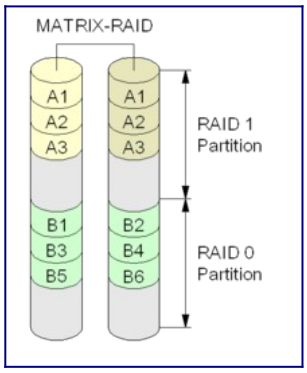


Diagrama de una configuración Matriz RAID

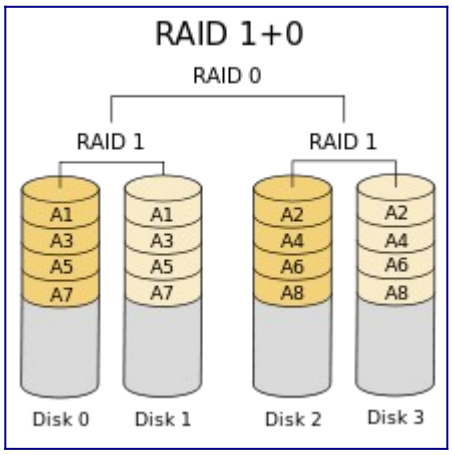
Matrix RAID (‘matriz RAID’) es una característica que apareció por vez primera en la BIOS RAID Intel ICH6R. No es un nuevo nivel RAID.

El Matrix RAID utiliza dos o más discos físicos, asignando partes de idéntico tamaño de cada uno de los diferentes niveles de RAID. Así, por ejemplo, sobre 4 discos de un total de 600GB, se pueden usar 200 en raid 0, 200 en raid 10 y 200 en raid 5. Actualmente, la mayoría de los otros productos RAID BIOS de gama baja solo permiten que un disco participen en un único conjunto.

Este producto está dirigido a los usuarios domésticos, proporcionando una zona segura (la sección

RAID 1) para documentos y otros archivos que se desean almacenar redundantemente y una zona más rápida (la sección RAID 0) para el sistema operativo, aplicaciones, etcétera.

Linux MD RAID 10



RAID 10

La controladora RAID software del kernel de Linux (llamada **md**, de *multiple disk*, 'disco múltiple') puede ser usada para construir un conjunto RAID 1+0 clásico, pero también permite un único nivel RAID 10 con algunas extensiones interesantes.3

En particular, soporta un espejado de *k* bloques en *n* unidades cuando *k* no es divisible por *n*. Esto se hace repitiendo cada bloque *k* veces al escribirlo en un conjunto RAID 0 subyacente de *n* unidades. Evidentemente esto equivale a la configuración RAID 10 estándar.

Linux también permite crear otras configuraciones RAID usando la controladora *md* (niveles 0, 1, 4, 5 y 6) además de otros usos no RAID como almacenamiento multirruta y LVM2.

IBM ServeRAID 1E

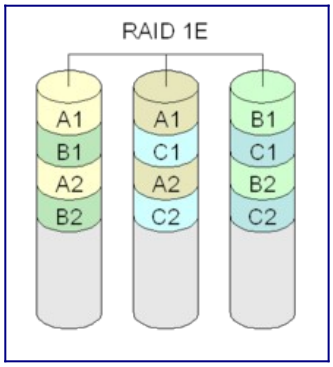


Diagrama de una configuración RAID 1E

La serie de adaptadores IBM ServeRAID soportan un espejado doble de un número arbitrario de discos, como se ilustra en el gráfico.

Esta configuración es tolerante de fallos de unidades no adyacentes. Otros sistemas de almacenamiento como el StorEdge T3 de Sun soportan también este modo.

RAID Z

El sistema de archivos ZFS de Sun Microsystems implementa un esquema de redundancia integrado parecido al RAID 5 que se denomina **RAID Z**. Esta configuración evita el «agujero de escritura» del RAID 56 y la necesidad de la secuencia leer-modificar-escribir para operaciones de escrituras pequeñas efectuando solo escrituras de divisiones (*stripes*) completas, espejando los bloques pequeños en lugar de protegerlos con el cálculo de paridad, lo que resulta posible gracias a que el sistema de archivos conoce la estructura de almacenamiento subyacente y puede gestionar el espacio adicional cuando lo necesita.

→ Infraestructura actual.

md-00 ~ # lsblk

```
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                 252:0    0   25G  0 disk
├─vda1              252:1    0    1G  0 part /boot
└─vda2              252:2    0   24G  0 part
  ├─rhel-pool00_tmeta   253:0    0   12M  0 lvm
  │ └─rhel-pool00-tpool 253:2    0 16,7G  0 lvm
  │   ├─rhel-root       253:3    0 16,7G  0 lvm  /
  │   └─rhel-pool00     253:5    0 16,7G  1 lvm
  ├─rhel-pool00_tdata   253:1    0 16,7G  0 lvm
  │ └─rhel-pool00-tpool 253:2    0 16,7G  0 lvm
  │   ├─rhel-root       253:3    0 16,7G  0 lvm  /
  │   └─rhel-pool00     253:5    0 16,7G  1 lvm
  └─rhel-swap         253:4    0  2,5G  0 lvm  [SWAP]
vdb                 252:16   0   20G  0 disk
vdc                 252:32   0   20G  0 disk
vdd                 252:48   0   20G  0 disk
vde                 252:64   0   20G  0 disk
```

vdf 252:80 0 20G 0 disk

md-00 ~ # man -k mdadm

mdadm (8) - manage MD devices aka Linux Software RAID

mdadm.conf (5) - configuration for management of Software RAID with mdadm

→ mdadm Básico.

mdadm -C --create

-l --nivel=

-S --stop

-n --raid-devices

-D

--scan

--fail

--remove

md-00 ~ # cat /proc/mdstat

Personalities :

unused devices: <none>

→ Niveles de RAID.

-l, --level=

Set RAID level. When used with --create, options are: linear, raid0, 0, stripe, raid1, 1, mirror, raid4, 4, raid5, 5, raid6, 6, raid10, 10, multipath, mp, faulty, container.

md-00 ~ # mdadm -Cv /dev/md10 --level=raid10 -n 4 /dev/vdb /dev/vdc /dev/vdd /dev/vde

mdadm: Defaulting to version 1.2 metadata

mdadm: array /dev/md10 started.

md-00 ~ # cat /proc/mdstat

Personalities : [raid10]

```
md10 : active raid10 vde[3] vdd[2] vdc[1] vdb[0]
      41908224 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
      [=========>.....]  resync = 48.2% (20222464/41908224) finish=1.7min
speed=207506K/sec
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10]
md10 : active raid10 vde[3] vdd[2] vdc[1] vdb[0]
      41908224 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
unused devices: <none>
```

md-00 ~ # mdadm -E /dev/vd[bcde] | grep Event

```
         Events : 17
         Events : 17
         Events : 17
         Events : 17
```

md-00 ~ # lsblk -f

```
NAME                MAJ:MIN RM  SIZE RO TYPE   MOUNTPOINT
…
vdb                 linux_ md-00.cadilinea.lan:10
│                                685869da-1c32-6721-1d9f-f79a8ec35a32
├─md10              xfs                ef1980d1-549d-447e-86d0-822cd2fb30ca
└─vdb1              linux_ md-00.cadilinea.lan:10
                                 685869da-1c32-6721-1d9f-f79a8ec35a32
vdc                 linux_ md-00.cadilinea.lan:10
│                                685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10              xfs                ef1980d1-549d-447e-86d0-822cd2fb30ca
vdd                 linux_ md-00.cadilinea.lan:10
│                                685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10              xfs                ef1980d1-549d-447e-86d0-822cd2fb30ca
```

vde linux_ md-00.cadilinea.lan:10

│ 685869da-1c32-6721-1d9f-f79a8ec35a32

└─md10 xfs ef1980d1-549d-447e-86d0-822cd2fb30ca

vdf

md-00 ~ # mkdir /mnt/raid10

md-00 ~ # ls /dev/md10 -alh

brw-rw----. 1 root disk 9, 10 mar 24 16:35 /dev/md10

---

md-00 ~ # fdisk /dev/md10

Bienvenido a fdisk (util-linux 2.32.1).

Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.

Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.

Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x994c6cbf.

Orden (m para obtener ayuda): p

Disco /dev/md10: 40 GiB, 42914021376 bytes, 83816448 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x994c6cbf

Orden (m para obtener ayuda): n

Tipo de partición

p primaria (0 primaria(s), 0 extendida(s), 4 libre(s))

e extendida (contenedor para particiones lógicas)

Seleccionar (valor predeterminado p): p

Número de partición (1-4, valor predeterminado 1):

Primer sector (2048-83816447, valor predeterminado 2048):

Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-83816447, valor predeterminado 83816447):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 40 GiB.

Orden (m para obtener ayuda): p

Disco /dev/md10: 40 GiB, 42914021376 bytes, 83816448 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x994c6cbf

Disposit. Inicio Comienzo Final Sectores Tamaño Id Tipo

/dev/md10p1 2048 83816447 83814400 40G 83 Linux

Orden (m para obtener ayuda): n

Tipo de partición

p primaria (0 primaria(s), 0 extendida(s), 4 libre(s))

e extendida (contenedor para particiones lógicas)

Seleccionar (valor predeterminado p): p

Número de partición (1-4, valor predeterminado 1):

Primer sector (2048-83816447, valor predeterminado 2048):

Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-83816447, valor predeterminado 83816447):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 40 GiB.

Orden (m para obtener ayuda): p

Disco /dev/md10: 40 GiB, 42914021376 bytes, 83816448 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x00e00ada

Disposit. Inicio Comienzo Final Sectores Tamaño Id Tipo

/dev/md10p1 2048 83816447 83814400 40G 83 Linux

Orden (m para obtener ayuda): t

Se ha seleccionado la partición 1

Código hexadecimal (escriba L para ver todos los códigos): 8e

Se ha cambiado el tipo de la partición 'Linux' a 'Linux LVM'.

Orden (m para obtener ayuda): p

Disco /dev/md10: 40 GiB, 42914021376 bytes, 83816448 sectores

Unidades: sectores de 1 * 512 = 512 bytes

Tamaño de sector (lógico/físico): 512 bytes / 512 bytes

Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes

Tipo de etiqueta de disco: dos

Identificador del disco: 0x00e00ada

Disposit. Inicio Comienzo Final Sectores Tamaño Id Tipo

/dev/md10p1 2048 83816447 83814400 40G 8e Linux LVM

rden (m para obtener ayuda): w

Se ha modificado la tabla de particiones.

Llamando a ioctl() para volver a leer la tabla de particiones.

Se están sincronizando los discos.

md-00 ~ # partprobe

md-00 ~ # mkfs.xfs /dev/md10p1

```
log stripe unit (524288 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/md10              isize=512    agcount=16, agsize=654720 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=10475520, imaxpct=25
         =                       sunit=128    swidth=256 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=5120, version=2
         =                       sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
```

```
md-00 ~ # lsblk  -f
NAME                FSTYPE          LABEL               UUID                                 MOUNTPOINT
…
vdb                 linux_ md-00.cadilinea.lan:10
│                                   685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10
  └─md10p1
                    xfs             e9197c61-d030-473a-a2b0-db26cab7973f
vdc                 linux_ md-00.cadilinea.lan:10
│                                   685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10
  └─md10p1
                    xfs             e9197c61-d030-473a-a2b0-db26cab7973f
vdd                 linux_ md-00.cadilinea.lan:10
```

```
│                   685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10
  └─md10p1
        xfs         e9197c61-d030-473a-a2b0-db26cab7973f
vde       linux_ md-00.cadilinea.lan:10
│                   685869da-1c32-6721-1d9f-f79a8ec35a32
└─md10
  └─md10p1
        xfs         e9197c61-d030-473a-a2b0-db26cab7973f
vdf
```

md-00 ~ # vim /etc/fstab

```
…
/dev/md10p1     /mnt/raid10   xfs    defaults      0      0
```

md-00 ~ # df -hT /dev/md10p1

```
S.ficheros     Tipo Tamaño Usados  Disp Uso% Montado en
/dev/md10p1    xfs    40G   319M   40G   1% /mnt/raid10
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10]
md10 : active raid10 vde[3] vdd[2] vdc[1] vdb[0]
      41908224 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
unused devices: <none>
```

→La Prueba, …

Actualmente 4 discos raid10 20 Gb. por disco → Total 40 GB|raid10.

md-00 ~ # touch /mnt/raid10/CrackDiscos.txt

md-00 ~ # echo "Hola raid10 - Tengo ahora 4 discos" >> /mnt/raid10/CrackDiscos.txt

→ Elimino un disco en caliente, …

md-00 ~ # lsblk

```
NAME                MAJ:MIN RM  SIZE RO TYPE   MOUNTPOINT
…
vdb                 252:16   0   20G  0 disk
└─md10                9:10   0   40G  0 raid10
  └─md10p1            259:0    0   40G  0 md     /mnt/raid10
vdc                 252:32   0   20G  0 disk
└─md10                9:10   0   40G  0 raid10
  └─md10p1            259:0    0   40G  0 md     /mnt/raid10
vdd                 252:48   0   20G  0 disk
└─md10                9:10   0   40G  0 raid10
  └─md10p1            259:0    0   40G  0 md     /mnt/raid10
vdf                 252:80   0   20G  0 disk
```

→ **Le tocó al 'e'.**

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10]
md10 : active raid10 vdd[2] vdc[1] vdb[0]
      41908224 blocks super 1.2 512K chunks 2 near-copies [4/3] [UUU_]
unused devices: <none>
```

md-00 ~ # cat /mnt/raid10/CrackDiscos.txt

```
Hola raid10 - Tengo ahora 4 discos → Mentira.
```

md-00 ~ # mdadm --detail /dev/md10

```
/dev/md10:
           Version : 1.2
     Creation Time : Thu Mar 25 10:25:50 2021
        Raid Level : raid10
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 4
```

Total Devices : 3

Persistence : Superblock is persistent

Update Time : Thu Mar 25 11:06:19 2021

State : clean, degraded

Active Devices : 3

Working Devices : 3

Failed Devices : 0

Spare Devices : 0

Layout : near=2

Chunk Size : 512K

Consistency Policy : resync

Name : md-00.cadilinea.lan:10 (local to host md-00.cadilinea.lan)

UUID : 685869da:1c326721:1d9ff79a:8ec35a32

Events : 24

Number	Major	Minor	RaidDevice	State	
0	252	16	0	active sync set-A	/dev/vdb
1	252	32	1	active sync set-B	/dev/vdc
2	252	48	2	active sync set-A	/dev/vdd
-	0	0	3	removed	

md-00 ~ # mdadm --monitor /dev/md10

Mar 25 11:17:24: DegradedArray on /dev/md10 unknown device

md-00 ~ # mdadm --stop /dev/md10

mdadm: Cannot get exclusive access to /dev/md10:Perhaps a running process, mounted filesystem or active volume group?

md-00 ~ # umount /mnt/raid10

[1574.554045] XFS (md10p1): Unmounting Filesystem

md-00 ~ # mdadm --stop /dev/md10

[1580.768392] md10: detected capacity change from 42914021376 to 0

[1580.769436] md: md10 stopped.

mdadm: stopped /dev/md10

md-00 ~ # mdadm --assemble --scan

mdadm: /dev/md/10 has been started with 3 drives (out of 4).

md-00 ~ # cat /proc/mdstat

Personalities : [raid10]

md10 : active raid10 vdb[0] vdd[2] vdc[1]

41908224 blocks super 1.2 512K chunks 2 near-copies [4/3] [UUU_]

unused devices: <none>

md-00 ~ # mdadm --manage /dev/md10 --add /dev/vde

mdadm: added /dev/vde

md-00 ~ # cat /proc/mdstat

Personalities : [raid10]

md10 : active raid10 vde[4] vdb[0] vdd[2] vdc[1]

41908224 blocks super 1.2 512K chunks 2 near-copies [4/3] [UUU_]

[=>.....] recovery = 7.2% (1529344/20954112) finish=5.2min speed=61173K/sec

md-00 ~ # mdadm --detail /dev/md10

/dev/md10:

Version : 1.2

Creation Time : Thu Mar 25 10:25:50 2021

Raid Level : raid10

```
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 4
     Total Devices : 4
       Persistence : Superblock is persistent
       Update Time : Fri Mar 26 11:19:32 2021
             State : clean, degraded, recovering
    Active Devices : 3
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 1
            Layout : near=2
        Chunk Size : 512K
Consistency Policy : resync
    Rebuild Status : 87% complete
              Name : md-00.cadilinea.lan:10  (local to host md-00.cadilinea.lan)
              UUID : 685869da:1c326721:1d9ff79a:8ec35a32
            Events : 63
    Number   Major   Minor   RaidDevice State
       0     252       16        0      active sync set-A   /dev/vdb
       1     252       32        1      active sync set-B   /dev/vdc
       2     252       48        2      active sync set-A   /dev/vdd
       4     252       64        3      spare rebuilding   /dev/vde
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10]
md10 : active raid10 vde[4] vdb[0] vdd[2] vdc[1]
      41908224 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
unused devices: <none>
```

md-00 ~ # mdadm --brief /dev/md10

/dev/md10: 39.97GiB raid10 4 devices, 0 spares. Use mdadm --detail for more detail.

md-00 ~ # mdadm --grow /dev/md10 -l 6 -n 4

mdadm: RAID10 can only be changed to RAID0

md-00 ~ # mdadm --grow /dev/md10 -l 0 -n 4

mdadm: New number of raid-devices impossible for RAID10

md-00 ~ # mdadm --grow /dev/md10 -l 0 -n 2

mdadm: level of /dev/md10 changed to raid0

md-00 ~ # mdadm --grow /dev/md10 -l 6 -n 4 --force

mdadm: /dev/md10: could not set level to raid6

md-00 ~ # cat /proc/mdstat

Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]

md10 : active raid0 vde[4] vdc[1]

41908224 blocks super 1.2 512k chunks

md-00 ~ # lsblk

```
NAME                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
…
vdb                 252:16   0   20G  0 disk
└─vdb1              252:17   0   20G  0 part
vdc                 252:32   0   20G  0 disk
└─md10                9:10   0   40G  0 raid0
  └─md10p1          259:0    0   40G  0 md
vdd                 252:48   0   20G  0 disk
vde                 252:64   0   20G  0 disk
└─md10                9:10   0   40G  0 raid0
```

```
└─md10p1            259:0    0   40G  0 md
```

md-00 ~ # mdadm --grow /dev/md10 -l 5

```
mdadm: level of /dev/md10 changed to raid5
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]
md10 : active raid5 vde[4] vdc[1]
      41908224 blocks super 1.2 level 5, 512k chunk, algorithm 5 [3/2] [UU_]
      [>.....................]  reshape =  4.9% (1034216/20954112) finish=15.7min speed=21106K/sec
unused devices: <none>
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]
md10 : active raid5 vde[4] vdc[1]
      41908224 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
unused devices: <none>
```

md-00 ~ # mdadm /dev/md10 --add-spare /dev/vdb

```
mdadm: added /dev/vdb
```

md-00 ~ # mdadm /dev/md10 --add-spare /dev/vdd

```
mdadm: added /dev/vdd
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]
md10 : active raid5 vdd[5](S) vdb[3] vde[4] vdc[1]
      41908224 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

md-00 ~ # lsblk

```
…
vdb                 252:16   0   20G  0 disk
├─md10              9:10   0   40G  0 raid5
```

```
│ └─md10p1          259:0    0   40G  0 md    /mnt/raid10
└─vdb1             252:17   0   20G  0 part
vdc                252:32   0   20G  0 disk
└─md10             9:10     0   40G  0 raid5
  └─md10p1         259:0    0   40G  0 md    /mnt/raid10
vdd                252:48   0   20G  0 disk
└─md10             9:10     0   40G  0 raid5
  └─md10p1         259:0    0   40G  0 md    /mnt/raid10
vde                252:64   0   20G  0 disk
└─md10             9:10     0   40G  0 raid5
  └─md10p1         259:0    0   40G  0 md    /mnt/raid10
```

md-00 ~ # mdadm --detail /dev/md10

```
/dev/md10:
           Version : 1.2
     Creation Time : Thu Mar 25 10:25:50 2021
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Fri Mar 26 19:21:07 2021
             State : clean
    Active Devices : 3
   Working Devices : 4
```

```
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : md-00.cadilinea.lan:10  (local to host md-00.cadilinea.lan)
              UUID : 685869da:1c326721:1d9ff79a:8ec35a32
            Events : 198

    Number   Major   Minor   RaidDevice State
       1     252       32        0      active sync   /dev/vdc
       4     252       64        1      active sync   /dev/vde
       3     252       16        2      active sync   /dev/vdb

       5     252       48        -      spare   /dev/vdd
```

md-00 ~ # mdadm --grow /dev/md10 -l 6

```
mdadm: level of /dev/md10 changed to raid6
```

md-00 ~ # cat /proc/mdstat

```
Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]
md10 : active raid6 vdd[5] vdb[3] vde[4] vdc[1]
      41908224 blocks super 1.2 level 6, 512k chunk, algorithm 18 [4/3] [UUU_]
      [>.....................]  reshape =  1.3% (274944/20954112) finish=73.9min speed=4660K/sec
unused devices: <none>
```

md-00 ~ # mdadm --detail /dev/md10

```
/dev/md10:
           Version : 1.2
     Creation Time : Thu Mar 25 10:25:50 2021
        Raid Level : raid6
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 4
     Total Devices : 4
       Persistence : Superblock is persistent
       Update Time : Fri Mar 26 21:19:33 2021
             State : clean
    Active Devices : 4
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 0
            Layout : left-symmetric
        Chunk Size : 512K
Consistency Policy : resync
              Name : md-00.cadilinea.lan:10  (local to host md-00.cadilinea.lan)
              UUID : 685869da:1c326721:1d9ff79a:8ec35a32
            Events : 467

    Number   Major   Minor   RaidDevice State
       1     252       32        0      active sync   /dev/vdc
       4     252       64        1      active sync   /dev/vde
       3     252       16        2      active sync   /dev/vdb
       5     252       48        3      active sync   /dev/vdd
```

→Añadimos otro disco:

md-00 ~ # lsblk

…

vde 252:64 0 20G 0 disk

└─md10 9:10 0 40G 0 raid6

└─md10p1 259:0 0 40G 0 md /mnt/raid10

vdf 252:80 0 20G 0 disk

md-00 ~ # mdadm /dev/md10 --add-spare /dev/vdf

mdadm: added /dev/vdf

md-00 ~ # mdadm --detail /dev/md10

…

Number	Major	Minor	RaidDevice	State	
6	252	80	0	spare rebuilding	/dev/vdf
4	252	64	1	active sync	/dev/vde
3	252	16	2	active sync	/dev/vdb
5	252	48	3	active sync	/dev/vdd
1	252	32	-	faulty	/dev/vdc

md-00 ~ # cat /proc/mdstat

Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]

md10 : active raid6 vdf[6](S) vdd[5] vdb[3] vde[4] vdc[1]

41908224 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]

unused devices: <none>

→Provocamos fallo en un disco:

md-00 ~ # mdadm /dev/md10 --fail /dev/vdc

mdadm: set /dev/vdc faulty in /dev/md10

md-00 ~ # cat /proc/mdstat

Personalities : [raid10] [raid0] [raid6] [raid5] [raid4]

```
md10 : active raid6 vdf[6] vdd[5] vdb[3] vde[4] vdc[1](F)
      41908224 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/3] [_UUU]
      [=>.....................]  recovery =  6.3% (1330300/20954112) finish=45.5min speed=7168K/sec
unused devices: <none>
```

md-00 ~ # mdadm /dev/md10 --remove /dev/vdc

```
mdadm: hot removed /dev/vdc from /dev/md10
```

md-00 ~ # mdadm--detail /dev/md10c

```
…
   Number   Major   Minor   RaidDevice State
      6     252      80        0      active sync   /dev/vdf
      4     252      64        1      active sync   /dev/vde
      3     252      16        2      active sync   /dev/vdb
      5     252      48        3      active sync   /dev/vdd
```

md-00 ~ # lsblk

```
…
vdb               252:16   0   20G  0 disk
├─md10              9:10   0   40G  0 raid6
│ └─md10p1          259:0    0   40G  0 md    /mnt/raid10
└─vdb1            252:17   0   20G  0 part
vdc               252:32   0   20G  0 disk
vdd               252:48   0   20G  0 disk
└─md10              9:10   0   40G  0 raid6
  └─md10p1          259:0    0   40G  0 md    /mnt/raid10
vde               252:64   0   20G  0 disk
└─md10              9:10   0   40G  0 raid6
  └─md10p1          259:0    0   40G  0 md    /mnt/raid10
```

```
vdf             252:80   0   20G  0 disk
└─md10            9:10   0   40G  0 raid6
  └─md10p1      259:0    0   40G  0 md    /mnt/raid10
```

BIBLIOGRAFIA:

https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/7/html/installation_guide/sect-installation-planning-partitioning-raid-ppc

https://hardzone.es/tutoriales/montaje/raid-discos-duros/

https://es.wikipedia.org/wiki/RAID

https://www.ducea.com/2009/03/08/mdadm-cheat-sheet/

https://raid.wiki.kernel.org/index.php/A_guide_to_mdadm